The International Journal of Digital Curation
Issue 1, Volume 5 | 2010

Towards Interoperable Preservation Repositories: TIPR

Priscilla Caplan,
Florida Center for Library Automation
William Kehoe,
Cornell University Library
Joseph Pawletko,
Bobst Library, New York University

Abstract

Towards Interoperable Preservation Repositories (TIPR) is a project funded by the Institute of Museum and Library Services to create and test a Repository eXchange Package (RXP). The package will make it possible to transfer complex digital objects between dissimilar preservation repositories. For reasons of redundancy, succession planning and software migration, repositories must be able to exchange copies of archival information packages with each other. Every different repository application, however, describes and structures its archival packages differently. Therefore each system produces dissemination packages that are rarely understandable or usable as submission packages by other repositories. The RXP is an answer to that mismatch. Other solutions for transferring packages between repositories focus either on transfers between repositories of the same type, such as DSpace-to-DSpace transfers, or on processes that rely on central translation services. Rather than build translators between many dissimilar repository types, the TIPR project has defined a standards-based package of metadata files that can act as an intermediary information package, the RXP, a lingua franca all repositories can read and write.[1]

[1] This article is based on the paper given by the authors at iPRES 2009; received January 2010, published June 2010.

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. ISSN: 1746-8256 The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre.

Introduction

Towards Interoperable Preservation Repositories (TIPR) is a two year project partnership between the Florida Center for Library Automation, Cornell University and New York University. The project is funded by the Institute of Museum and Library Services and runs from October 2008 through September 2010. The goal of the project is to develop, test and promote a standard interchange format for exchanging stored information packages among OAIS-based preservation repositories.

There is wide agreement in the international preservation community that responsibility for long-term preservation of scientific and cultural heritage materials must be shared among many organizations. No single institution has the capacity to preserve all or even a significant portion of a nation or region's digital output. This awareness is evidenced in the structure of major preservation efforts internationally. For example, the goal of the congressionally funded National Digital Information Infrastructure and Preservation Program (NDIIPP) is not to build a federal preservation repository, but to build a national network of preservation partners who will collectively ensure access to content (Smith, 2006). Similarly Digital Preservation Europe, noting on its website that "[d]igital preservation is too big an issue for individual institutions or even sectors to address independently", has the mission of fostering collaboration and synergies between national initiatives in Europe.[2]

The model of distributed digital preservation dictates that there are and will continue to be many independent preservation repositories storing different content. Less obviously but equally compellingly, in some cases multiple repositories will store different versions of the same content. This is not simply a matter of having copies in different geographic locations. For materials of high value, it will only be prudent to "hedge one's bets" by entrusting them to multiple repositories managed by different organizations, running different software and using different preservation strategies. Especially now, when no institution has a long-term track record of successfully maintaining usable copies, it makes sense to err on the side of caution.

Use cases for transfer

Another implication is that there are several practical use cases for transferring copies of stored information packages from one repository to another. One case that will surely arise is for a custodial institution to take advantage of newer, more sophisticated or more specialized preservation options. At this time there are few true preservation repositories and most are operated for the use of particular constituencies. Imagine a library that archives its digital content in the only repository available to it, say a university-operated institutional repository. A few years later a third-party repository specifically tailored to Geographic Information Systems (GIS) opens for business. The library may want to deposit a copy of its archived GIS content in the special GIS repository.

A second use case involves succession planning. Imagine the institutional repository posited above is discontinued, perhaps because the university has an opportunity to become a member of a larger shared repository with improved preservation functionality. In this case the entire stored content of the institutional repository must be transferred to the shared repository. More generally, all

[2] DPE: About Digital Preservation Europe: http://www.digitalpreservationeurope.eu/about/

preservation repositories should have a plan for what should take place if they cease operation. The TRAC checklist item A1.2 requires a repository to have "... an appropriate, formal succession plan, contingency plans, and/or escrow arrangements in place in case the repository ceases to operate or the governing or funding institution substantially changes its scope." The transfer of content to one or more other repositories may in many cases be the preferred course of action (Trustworthy Repositories, 2007).

In a third case, an institution might decide to migrate from the system it is running to a more functional or more modern preservation repository system. In this case again the entire content of the old repository must be transferred to the new preservation application.

Approaches to the problem

A number of initiatives implement what they call "distributed" digital preservation, where copies of a repository's content are stored in multiple locations. This can be implemented with grid storage or with application-level replication as in private LOCKSS (Lots Of Copies Keeps Stuff Safe) networks (Halbert, 2009; Moore, 2006; Reich & Rosenthal, 2009). These approaches, however, rely on the creation and ongoing verification of identical copies of content. They do not address the use cases in the Introduction, where the content must be ingested into a different preservation repository system. Different applications use different packaging, require different processing information, supply different metadata, and may implement different preservation strategies. Transferring content from one application to another is not a simple matter of making exact copies in different storage locations.

According to the Open Archival Information Systems (OAIS) reference model (Consultative Committee on Space Data Systems, 2002), digital objects are submitted to preservation repositories in Submission Information Packages (SIP). The process of Ingest transforms a SIP into an Archival Information Package (AIP) for storage, and the process of dissemination transforms an AIP into a Dissemination Information Package (DIP) for external use. In OAIS terms, transferring a copy of a stored object from repository A to repository B is a matter of A transforming a stored AIP into a DIP sent to B for ingest. However, because different repository systems describe and structure their archival packages differently, a DIP produced by one repository is unlikely to be directly usable as a SIP by another. Therefore A's DIP must somehow be transformed into a SIP that B can ingest.

Of course, if each repository system's DIP format is mapped to every possible exchange partner's SIP format, the number of mappings required for any significant number of repositories would be prohibitive. Clearly a single common exchange format is called for, so that each repository requires only two mappings: one from its own DIP to the exchange format and one from the exchange format to its own SIP.

The ECHO DEPository, an NDIIPP-funded research project, implemented this solution in its Hub and Spoke Framework Tool Suite (HandS) (Habing, Eke, Cordial, Ingram & Manaster, 2009). In the HandS architecture, a central Hub service provides the tools to do the mappings between the package formats native to each repository application and a common exchange format, called the "Hub Package". As part of this conversion, the service augments the new Hub Package with several types of metadata:

a MODS (Metadata Object Description Schema) descriptive metadata record, derived technical metadata for each file in the package, and when possible, PREMIS (Preservation Metadata: Implementation Strategies) Event records documenting changes to the package. The central service also provides a "client" program to initiate the act of pulling a SIP from a repository and pushing a DIP to it. The client invokes a service called "LRCRUD" written specifically for each repository to interact with the repository's native interface.

The incentive for HandS came out of another initiative to evaluate and compare commonly used institutional and preservation repository systems. That project quickly ascertained that the systems had minimal interoperability and minimal support for active preservation strategies to compensate for format obsolescence. As a consequence, the HandS architecture was shaped by not one but by three quite distinct objectives: to support interoperability among the repository systems, to allow a central organization to control exchanges between repositories, and to provide essential preservation metadata not provided by the repository system.

Outside of a project- or research-based context, however, this approach has disadvantages. Success depends on the existence and efficacy of the central Hub service, which presumably must either be maintained by some organization or become a community resource, each of which has its own perils. Some software components must be maintained centrally and some at each repository's site. The Hub is responsible for initiating transfers via the client, taking this out of the repository's hands. The Hub service does an extensive amount of metadata creation (technical metadata for files, PREMIS Events) and metadata conversion (from the repository's format to Aquifer MODS) which is notoriously error-prone and sensitive to updated schema and new versions of tools such as JHOVE (JSTOR/Harvard Object Validation Environment).

The project Toward Interoperable Preservation Repositories (TIPR) also defines a common exchange package format based on PREMIS and METS (Metadata Encoding and Transmission Standard), but takes a more lightweight approach with no Hub services. TIPR has a single objective, to devise a standard method to enable content from one OAIS-based preservation repository to be ingested by another OAIS-based repository. In the real-world use cases described above, details of the transfer need to be a matter of agreement between the partner repositories, with all aspects under the full control of the partners. Dependence on an externally maintained central service would be both unnecessary and cumbersome. Moreover, participants in the TIPR project believe that improving the preservation functionality of repository applications is a problem best addressed by the developers of these systems. Institutional repositories can be retooled to support full digital preservation, or other applications designed specifically for long-term preservation can be used such as Ex Libris's Rosetta, Tessela's Storage Deposit Box, or the open source DAITSS (Dark Archive In The Sunshine State). The place to address shortcomings in preservation functionality is not in the process of exchange.

TIPR defines a common exchange package format, the Repository Exchange Package (RXP), and leaves it to the developers of each repository application to develop its own mappings to and from the RXP. Conversion, however, is a relatively simple affair, since package creation does not involve any conversion or creation of descriptive or technical metadata. The TIPR model is shown in Figure 1.

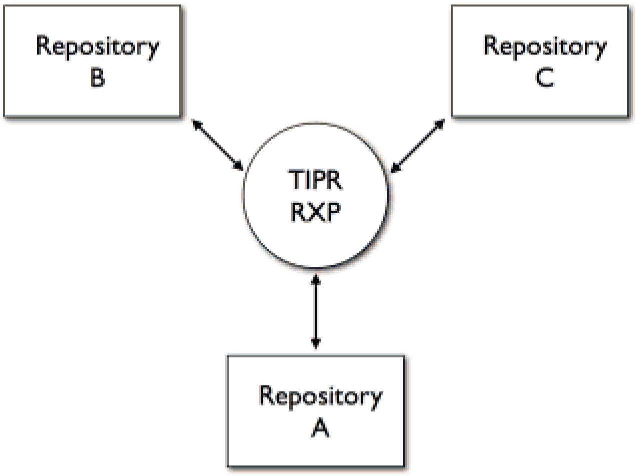

Figure 1. The TIPR model of package exchange.

Design Issues

The RXP was designed to meet the following requirements:

1) The exchange package must use existing standards well known to the cultural heritage preservation community. The project did not want to burden the community with yet another "standard" conflicting or overlapping with standards already in use.
2) The exchange package must be flexible enough to accommodate any repository's AIP; that is, it must be agnostic to the internal structure of the AIP.
3) The exchange package must contain enough information for the target repository to know what it is receiving at both the package level and the representation level.
4) Some selected information provided by the sending repository must actually be understood by the receiving repository.

Because of requirement (1), the TIPR RXP, like the Hub package, is based upon METS and PREMIS. TIPR assumes that the preservation community knows how to interpret METS syntax and PREMIS semantics. These two standards represent the core of a meaningful exchange.

Requirements (2) and (3) mean that any repository system can use the RXP no matter what their treatment of representations with no change to their internal AIP architectures. In the PREMIS data model, a representation is defined as the set of files needed to fully render an intellectual entity. For example, a particular journal article might consist of 14 data files: an XML file, a stylesheet, and a dozen images. An alternate representation of the same article might consist of a single PDF file. Depending on the repository architecture, an archive could store the two representations in a single AIP or in two AIPs. If the images were migrated from JPEG to JPEG2000, a third representation would be created, and again, a given repository could treat the three representations as one, two or three AIPs.

More importantly, preservation systems implementing format normalization or forward migration may execute different policies concerning which representations are retained after a transformation. If a source file goes through successive migrations (e.g. Corel WordPerfect to Microsoft Office Word 2003 to Open Office ODF) repository A may retain all three versions, repository B may retain the latest version only, and repository C may retain the original and latest only, discarding intermediates. A receiving repository, therefore, must be able to distinguish multiple representations within an arriving package. The three TIPR project partners have implemented preservation repositories with quite different approaches to representations, so they provide a good testbed for these requirements.

Requirement (4) dictates that certain information critical to digital preservation must be not only stored but also understood by the target (receiving) repository. "Understanding" in this context means that the metadata elements and values can be mapped to equivalent elements and values in the receiving system. A meaningful exchange in the preservation context dictates that interoperability must be semantic as well as syntactic.

A major early task of the project was to decide which types of metadata potentially contributed by the sending repository would need to be understood by the receiving repository. Various categories of administrative, preservation, and format-specific technical metadata were analyzed in turn. The project decided that most types of metadata could be recreated by the receiving repository, simply stored as received, or covered by a repository-to-repository service agreement. However, information pertaining to rights and to digital provenance (the history of ownership and actions affecting the object) must be understood.

The case for rights metadata is straightforward, since actionable rights information may control what access to the object is allowed and what preservation actions can be performed. If rights are the same for all materials transferred, they can be specified in a service agreement, but if they vary from package to package, each package must contain its own rights information.

The case for digital provenance information is also compelling. An OAIS-based preservation repository will perform many actions on a SIP in order to transform it into an AIP, which may or may not include creating transformed versions of source files. A common preservation strategy for both libraries and archives is to guard against format obsolescence by reformatting archived content files. A normalized version of a source file may be created in a format considered to be more preservation-worthy (stable, well understood, non-proprietary, etc.). A migrated version may be created in a more current version of the format, or a successor format. The original source file may be retained in archival storage or discarded in favor of the derivative version(s). In a preservation environment in which transformations may occur, the only way to guarantee the continued authenticity of the digital object is to maintain an unbroken record of digital provenance.

In PREMIS, rights and permissions relevant to the preservation of the object are described by the Rights entity, and digital provenance is described by the Events entity. Both Rights and Events can be associated with Agents, which can be persons, organizations or software. TIPR uses PREMIS as a meta-language for expressing these concepts regardless of the way they are represented in the sending or receiving repository system.

The Repository Exchange Package (RXP)

The RXP specification is available on the TIPR project site[3]. A minimal RXP for a package containing a single representation consists of exactly five required XML documents and a directory of files from the sending repository's AIP. An optional signature file can also be included as shown in Figure 2.

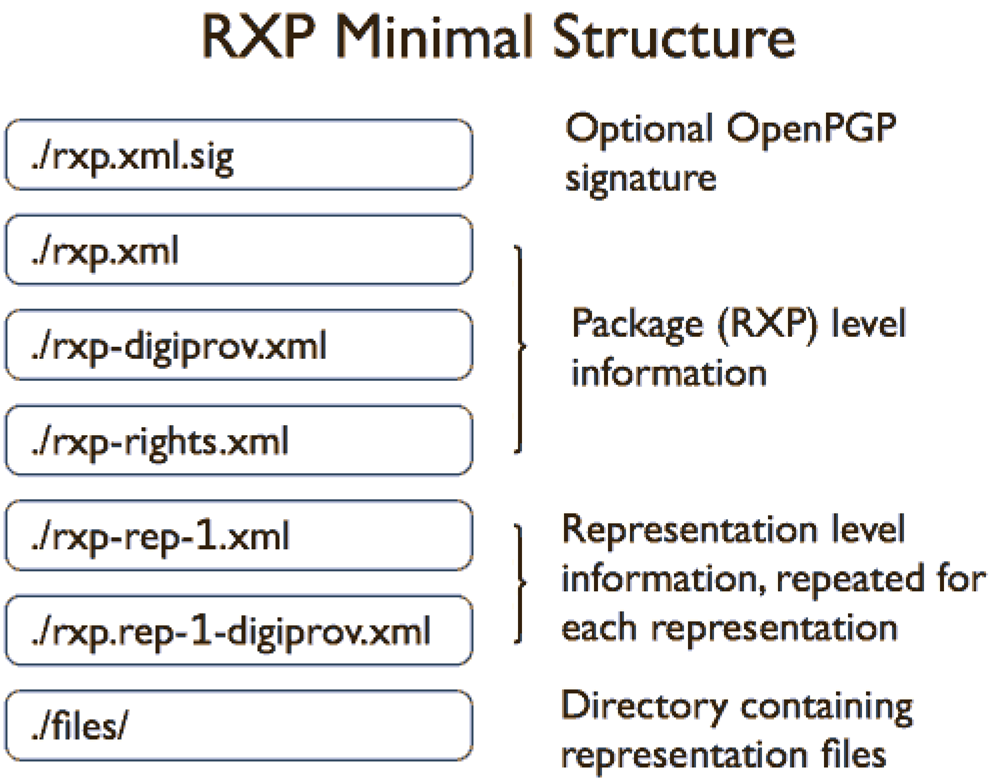

Figure 2. RXP Minimal Structure.

The three XML documents rxp.xml, rxp-digiprov.xml, and rxp-rights.xml contain information about the exchange package itself. rxp.xml is a METS document containing basic information about the package as a whole: the name of the package, the name of the sending repository, and the identity of the "active" representation (see "Transfer Issues" below). It uses METS mdRef elements to point to rxp-digiprov.xml, rxp-rights.xml, and any rxp-rep-n.xml documents contained in the package.

rxp-digiprov.xml is a PREMIS document containing digital provenance (Event) information pertaining to the RXP package itself. rxp-rights.xml is a PREMIS document with package level Rights information. Optionally, a fourth package level file may be present, rxp.xml.sig, containing a digital signature in OpenPGP format generated using the sender's private key and rxp.xml. The key exchange protocol is left up to the exchanging repositories.

[3] Towards Interoperable Preservation Repositories (TIPR): http://wiki.fcla.edu:8000/TIPR

The remaining two files in Figure 2, rxp-rep-1.xml and rxp-rep-1-digiprov.xml, describe the first (possibly only) representation in the package. rxp-rep-1.xml is a METS document with a <fileSec> and flat <structMap> identifying the files comprising the representation. rxp-rep-1-digiprov.xml is a PREMIS document containing, at a minimum, digital provenance information for that representation. This pair of files should be repeated for every representation n in the AIP, as rxp-rep-n.xml and rxp-rep-n-digiprov.xml. Each rxp-rep-n.xml contains the pointer to its corresponding rxp-rep-n-digiprov.xml.

The files directory contains the actual files comprising the AIP; that is, all of the files in all of the representations identified in rxp.xml.

The Transfer Tests

The RXP format is being tested in a series of inter-repository exchanges to ascertain the extent to which the partner systems can send and receive packages with minimal loss of information and maximum understanding. The project is unconcerned with the mechanics of file transfer, which would in reality be negotiated between repositories and specified in a Service Agreement. For the purpose of the project, test RXP packages were bundled according to the BagIt specification (Boyko, Kunze, Littman, Madden, & Vargas, 2009) and transferred by HTTP.

The first round of testing consisted of bilateral transfers. Validation scripts using Schematron (JTC1/SC34, 2006) were written to validate the five XML document types in the RXP. Each partner created, validated, and sent two RXP packages to each of the other partners to ingest. The main purpose of this testing was to help develop the draft of the RXP specification by exercising it in practice. The tests were repeated many times as the specification was tweaked, and as the partners refined their ideas about what information needed to be passed from sending to receiving repositories.

The bilateral transfers proved the feasibility of creating and processing the RXP, and gave some indication of the amount of effort it would require of repository developers and/or implementers. The trial showed that it was only a moderate amount of work to produce an RXP, because in all cases the repository's native SIP could be converted into an RXP using stylesheet transformations and scripts. Ingesting a foreign RXP into a repository proved to be more work, as merely transforming the RXP into the format of a native SIP did not ensure a proper ingest, and repository code actually had to be changed. In the case of the Florida Digital Archive, for example, the DAITSS software did not expect a SIP to contain externally-created provenance information and did not process it properly. Ultimately, however, all three partner repositories were made capable of ingesting an RXP.

The second set of tests were round trip transfers, where repository A sent an RXP to repository B; repository B ingested the package and exported it as an RXP back to repository A; and repository A re-ingested the package and compared the resulting AIP to the original AIP in the chain. These tests raised policy as well as technical issues. What should happen, for example, if repository A has a policy of always retaining original source files and repository B does not? In the real world, of course, the parties involved in a transfer would need to understand each other's policies and requirements, and would decide what to do on a case by case basis.

Evaluation of round-trip results was challenging, as it called for the sending partner to try to interpret the receiving partner's interpretation of the original RXP. The sending partner had to analyze how the returned package related to the source AIP, and whether any files that had to be preserved were missing. Part of the intent of this test was to demonstrate that an RXP could be deciphered by a receiving repository to the sender's level of satisfaction. It became evident that the bar for satisfaction would be different depending on the use case presumed for the transfer. Where the motive was diversification (wanting valuable content stored in multiple, heterogeneous repositories), understanding the metadata provided by the sender was less important than storing it as a preservation object. Where the motive was succession planning or system migration, it was important that certain metadata could be interpreted and processed appropriately.

The next set of tests to be undertaken will again be round-trip transfers with a larger set of RXPs, some of which have been intentionally edited to contain errors. The purpose of this is to identify flaws in the tools that implement the specification. Errors not exposed by the sender show problems on the sending end, while those not caught by the receiving repository show flaws on the receiving end. The testing may also generate suggestions for handling errors which can be included in guidelines or best practices documentation.

Transfer issues

Preserving information such as package names and file names through multiple transfers has been a challenge. For example, the initial plan was to record the original and current RXP identifiers as attributes in the root METS element of rxp.xml. This breaks as soon as the package endures a third transfer. This led to the definition of rxp-digiprov.xml which can track the full history of a particular RXP.

Each institution's repository has its own method for identifying objects, events, and agents. To avoid potential conflicts between these identifiers when transferring digital provenance, the RXP specification currently requires all PREMIS identifiers to be URIs. This would make identifiers universally exchangeable between any number of repositories. The requirement is being reconsidered, however, since the value of a METS ID attribute can not be a URI and this would force the use of two different identifier schemes.

Since different repositories implement different preservation strategies, the number of representations that they keep of a given intellectual entity, and the intended use of those representations, can vary widely. For example, one system may do successive forward migrations based on the original source representation only, while another will do each successive migration on the current migrated version at the time. Various representations may be considered by the source repository to be most authoritative, current, and/or usable; to be the source of future action; and/or to be included in default disseminations. Because of this variation, the TIPR project did not pursue a universal way to indicate the function or treatment status of a representation but left these matters to the repository service agreement. For example, the project did define a marker for the "active" representation, but the interpretation of this marker is left up to agreement between sending and receiving repositories.

The project exposed a limitation of the PREMIS Data Dictionary. Receiving repositories need some description of the exchange package itself, including its provenance and what high level rights adhere to it. PREMIS is capable of describing this, but the highest level of description in PREMIS is a representation object. An RXP package can contain multiple representations, and is more comparable to an Intellectual Entity than to anything else in the PREMIS data model. Unfortunately, the Intellectual Entity is out of scope for PREMIS as currently conceived. The TIPR project has asked the PREMIS Editorial Committee to consider allowing PREMIS elements describe intellectual entities when applicable, and the Committee has this on their agenda. In the meantime, the project uses PREMIS to describe the RXP as a whole in violation (or extension) of that standard.

Two larger issues concerned not the mechanics of repository-to-repository transfer but the broader context in which this takes place. An early concern involved rights. The project participants were uncertain whether, under international copyright law, there were two sets of rights that had to be taken into account: 1) the right and permissions pertaining to the archived content itself, and 2) rights that the sending repository might have in the metadata and packaging information it created for the AIP. The draft RXP accommodated the first but not the second. Consultation with an expert on copyright who also has some involvement with digital preservation indicated that the second case would probably not apply. An RXP and underlying AIP could be considered compilations of existing information, which are protected by copyright only if creativity is exercised in their assembly. We could argue comfortably that applications producing packages according to RXP or AIP specifications were not exercising creativity. The metadata contained within the package might be considered a derivative work, which is protected if there is added authorship or judgment but not protected to the extent it is only facts. While there is some room for judgment in, say, assigning a format designation, it seems that most technical and preservation metadata records facts, and therefore would not be subject to copyright. As a result, the RXP does not provide a place for recording the second type of rights information.

The other broad issue concerns the amount of information an exchange package can reasonably be expected to communicate, which is limited. It is evident that successful exchange will require service agreements between participating repositories to supplement the information contained in the RXP, so a model service agreement is being drafted as part of the project. At a minimum, it will have to document the following:

1) details of RXP composition by the source repository in this particular transfer, where the RXP specification allows options;
2) how the RXP will be transferred from source to target repository;
3) actions to be performed by the target repository on receipt of the RXP;
4) rights and restrictions on content not expressed within the exchange package;
5) archiving and preservation treatment of the ingested RXP by the target repository including which events must be recorded;
6) financial arrangements between source and target repositories;
7) legal aspects of the arrangement.

The first three items are quasi-technical, assuring the mechanics of transfer are addressed. Stipulations concerning (3), for example, would include what acknowledgements the sending repository can expect to get from the target repository.

Item (4) is necessary even if rights are communicated in the PREMIS Rights entity, because the receiving repository may or may not have the capability to implement all restrictions exactly or to take action on encoded rights information. The last three items are of critical concern to the owners of the content, and might influence decisions such as succession planning.

Conclusion

The TIPR RXP is not expected to be the final solution to the problem of repository interoperability, but it is intended to move the problem space out of a pure research environment into the realm of the operational requirements of production preservation repositories. The working version of the RXP specification, tools to validate the RXP, and project results and reports are available on the TIPR website.[4]

Acknowledgements

The TIPR project is supported by a two-year grant from the Institute of Museum and Library Services.

The authors thank Francesco Lazzarino and Marly Wilson for their help in drafting this paper.

References

Boyko, A., Kunze, J., Littman, J., Madden, L., Vargas, B. (2009). *The BagIt File Packaging Format* (v0.96). Retrieved January 19, 2010, from http://www.digitalpreservation.gov/library/resources/tools/docs/bagitspec.pdf

Consultative Committee on Space Data Systems. (2002, January). *Reference Model for an Open Archival Information System (OAIS)*. CCSDS 650.0-B-1: Blue Book (1). Retrieved January 8, 2010, from http://public.ccsds.org/publications/archive/650x0b1.pdf

Habing, T., Eke, J., Cordial, M., Ingram, W., & Manaster, R. (2009). Developments in digital preservation at the University of Illinois: The hub and spoke architecture for supporting repository interoperability and emerging preservation standards. *Library Trends, 57*(3). Retrieved January 9, 2010, from http://muse.jhu.edu/journals/library_trends/v057/57.3.habing.pdf

Halbert, M. (2009). Comparison of strategies and policies for building distributed digital preservation infrastructure: Initial findings from the MetaArchive Cooperative. *International Journal of Digital Curation, 4*(2). Retrieved January 11, 2010 from http://www.ijdc.net/index.php/ijdc/article/view/117

JTC1/SC34 (2006). Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation – Schematron. ISO/IEC 19757-3:2006.

[4] Towards Interoperable Preservation Repositories (TIPR): http://wiki.fcla.edu:8000/TIPR

Moore, R. W. (2006). Building preservation environments with data grid technology. *American Archivist, 19*(1). Retrieved January 11, 2010, from http://archivists.metapress.com/content/176p51l2w5278567/fulltext.pdf

Reich, V. & Rosenthal, D. (2009). Distributed digital preservation: Private LOCKSS networks as business, social, and technical frameworks. *Library Trends, 57*(3). Retrieved January 11, 2010, from http://muse.jhu.edu/journals/library_trends/v057/57.3.reich.pdf

Smith, A. (2006). Distributed preservation in a national context: NDIIPP at mid-point. *D-Lib Magazine, 12*(6). Retrieved January 8, 2010, from http://www.dlib.org/dlib/june06/smith/06smith.html

Trustworthy Repositories Audit & Certification: Criteria and Checklist (TRAC). (2007). Retrieved January 9, 2010, from http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf